



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/077,083	02/15/2002	Thomas E. Hamilton	09266.0006-00000	3495
22852	7590	01/26/2006	EXAMINER	
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP 901 NEW YORK AVENUE, NW WASHINGTON, DC 20001-4413			NANO, SARGON N	
			ART UNIT	PAPER NUMBER
			2157	

DATE MAILED: 01/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/077,083	HAMILTON ET AL.
Examiner	Sargon N. Nano	Art Unit 2157

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE ____ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 14 December 2005.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1 - 31, 33 - 40 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1 - 31, 33 - 40 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 7/03, 8/03, 9/03.
4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. ____.
5) Notice of Informal Patent Application (PTO-152)
6) Other: ____.

DETAILED ACTION

1. This office action is responsive to election/restriction mailed out on Nov. 7, 2005. Applicant's election without traverse of group 1 which consists of claims 1 – 31, 33 - 40 in the reply filed on Nov. 7, 2005 is acknowledged. Restriction is made **FINAL**.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 27 recites the limitation "“ a process hosted on the first computer or the second computer” There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1 – 31 and 33 – 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dandrea U.S. Patent No.6, 691,208 (referred to hereafter as Dandrea) in view of Bruno et al. U.S. Patent No. 6,725,456 (referred to hereafter as Bruno).

Dandrea teaches queuing architecture including a plurality of queues and associated method for controlling admission for disk access requests for video content (see abstract).

As to claim 1, Dandrea teaches a method of managing messages, comprising: storing messages in a plurality of queues (see col. 11 lines 15 – 33, Dandrea discloses a request entered into the queues);

providing a macro queue associated with the plurality of queues(see col. 14 line 33 – 47 and fig. 2 Dandrea discloses a queue selector for selecting queues);

calling an application programming interface to initiate a request to the macro queue to obtain a message stored in one of the plurality of queues without identifying a particular queue (see col. 9 lines 14 – 40, Dandrea discloses a schedule algorithm that is executed by the server) ; and

selecting a queue from among the plurality of queues and selecting a message from the selected queue(see col. 4 line 33 – 65 , Dandrea discloses a selected queue among a plurality of queues to be serviced).

Dandrea teaches the invention as mentioned above, Dandrea does not explicitly teaches an application programming interface, however Bruno teaches an application programming interface that provides an efficient generation and control or resource reservation on an operating system of a computer system, It would have been obvious to one of the ordinary skill in the art to include Application programming interface (API) in Dandrea invention because doing so would allow the language and the message

format used by an application program to communicate with a program that provides services for it.

As to claim 2, Dandrea teaches the method of claim 1, further comprising assigning a priority value to each of the plurality of queues (see col. 7 line 41 – col. 8, line 30).

As to claim 3, Dandrea teaches the method of claim 2 wherein the macro queue selects a message from a queue having the highest priority value (see col. 7 line 41 – col. 8, line 30).

As to claim 4, Dandrea teaches the method of claim 1 wherein the macro queue selects a message that has been stored in the plurality of queues for the longest time (see col. 4 line 48 – col. 5 line 40).

As to claim 5, Dandrea teaches the method of claim 1, further comprising providing a remote queue proxy for establishing a communication link between a remote application programming interface and the macro queue (see col. 10 line 46 – col. 11 line 2).

As to claim 6, Dandrea teaches the method of claim 1 wherein the plurality of queues and the macro queue are software objects that are implemented using object oriented programming principles (see col. 6 line 25 – 40).

As to claim 7, Dandrea teaches the method of claim 6, further comprising calling a software function of the macro queue object to associate a queue object with the macro queue object, wherein the software function returns a queue instance pointer

pointing to the location of the queue object and a priority value representing the priority of the queue (see col. 16 lines 15 – 57).

As to claim 8, Dandrea teaches the method of claim 6, further comprising calling a software function of the macro queue object to remove the association between the macro queue and a queue (see col. 6 line 25 – 40).

As to claim 9, Dandrea teaches a method of managing messages, comprising: providing an application programming interface (API) to allow a producer module to send a message to a macro queue that manages a plurality of queues, the API sending the message to the macro queue without identifying one of the plurality of queues (see col. 11 line 15 – 33).

Dandrea does not explicitly teach an application programming interface, however Bruno teaches an application programming interface that provides an efficient generation and control or resource reservation on an operating system of a computer system. It would have been obvious to one of the ordinary skill in the art to include Application programming interface (API) in Dandrea invention because doing so would allow the language and the message format used by an application program to communicate with a program that provides services for it.

As to claim 10, Dandrea teaches the method of claim 9 wherein the macro queue selects the first queue that is available among the plurality of queues and sends the message to the selected queue (see col. 4 lines 33 – 47 and fig. 2).

As to claim 11, Dandrea teaches the method of claim 9 wherein the macro queue duplicates the message and sends the message to all of the plurality of queues (see col. 5 line 61 - col. 6 line 56).

As to claim 12, Dandrea teaches the method of claim 9 wherein the macro queue selects a queue from among the plurality of queues that has the fastest response time based on previous response time records and sends the message to the selected queue (see col. 9 lines 50 – 57).

As to claim 13, Dandrea teaches the method of claim 9 wherein the macro queue selects a queue by cycling through each of the plurality of queues in a round robin fashion, and sends the message to the selected queue (see col. 9 lines 50 – 57).

As to claim 14, Dandrea teaches the method of claim 9 wherein the macro queue and the plurality of queues are implemented as software objects according to object oriented programming principles (see col. 6 lines 25 – 40).

As to claim 15, Dandrea teaches a method comprising:
keeping a list of queue pointers, each pointer pointing to one of a plurality of queues (see col. 10 lines 45 – col. 11 line 32);
receiving a request for adding a queue element (see col. 2 lines 23 – 46); and
servicing the request by selecting one or more queue pointers from the list based on a predetermined criterion and adding the queue element to the one or more queues that the selected one or more queue pointers are pointing to (see col. 2 lines 47 – col. 3 line 10).

As to claim 16, Dandrea teaches the method of claim 15 wherein the predetermined criterion is to select a queue pointer pointing to a queue that has the shortest response time (see col4 line 66 – col. 5 line 60).

As to claim 17, Dandrea teaches the method of claim 15 wherein the predetermined criterion is to select all of the queue pointers (see col4 line 66 – col. 5 line 60).

As to claim 18, Dandrea teaches the method of claim 15 wherein the predetermined criterion is to select a queue pointer from the list in a round robin fashion by cycling through each of the queue pointers in the list (see col.8 lines 31 – 45 and fig. 5).

As to claim 19, Dandrea teaches a method comprising:
keeping a list of queue pointers, each pointer pointing to one of a plurality of queues (see col. 10 line 45 – col. 11 line 33);

receiving a request for retrieving a queue element (see col. 2 lines 23 – 46); and servicing the request by selecting one or more queue pointers from the list based on a predetermined criterion and retrieving a queue element from the one or more queues that the selected one or more queue pointers are pointing to (see col. 2 line 47 – col. 3 line 10) . Dandrea does not explicitly teach an application programming interface, however Bruno teaches an application programming interface that provides an efficient generation and control or resource reservation on an operating system of a computer system, It would have been obvious to one of the ordinary skill in the art to include Application programming interface (API) in Dandrea invention because doing so would

allow the language and the message format used by an application program to communicate with a program that provides services for it.

As to claim 20, Dandrea teaches the method of claim 19 wherein the predetermined criterion is to select a queue pointer pointing to a queue that is the first one to be available (see col. 4 line 66 – col. 5 line 60).

As to claim 21, Dandrea teaches the method of claim 19 wherein each of the queues has a priority value, and the predetermined criterion is to select a queue pointers pointing to a queue having the highest priority value (see col.4 line 48 – col. 5 line 14).

As to claim 22, Dandrea teaches a method for messages communication in a distributed system, comprising:

providing an application programming interface on each computer of a group of computers in the distributed system (see col. 10 line 17 – 45 and fig. 1);

providing a remote queue proxy on each of the computers of the group (see fig. 1 Dandrea discloses video server);

initiating a request through an application programming interface on a first computer of the group (see col. 2 lines 23 – 46); and

passing said request to a second of the computers of the group by passing said request through the remote queue proxy on the first computer and the remote member queue proxy on said second computer(see fig. 1 Dandrea discloses a video server as a proxy and the disks as computers). Dandrea does not explicitly teaches an application programming interface, however Bruno teaches an application programming interface

that provides an efficient generation and control or resource reservation on an operating system of a computer system, It would have been obvious to one of the ordinary skill in the art to include Application programming interface (API) in Dandrea invention because doing so would allow the language and the message format used by an application program to communicate with a program that provides services for it.

As to claim 23, Dandrea teaches the method of claim 22 wherein providing the application programming interface includes providing software objects implementing said interface that are implemented using object oriented programming principles (see col. 6, lines 25 – 40).

As to claim 24, Dandrea teaches method of claim 22 wherein providing the remote queue proxy includes providing a software object implementing said proxy (see fig. 1 Dandrea discloses a video server as a proxy and the disks as computers).

As to claim 25, Dandrea teaches a method for passing messages between processes in a distributed system comprising:

providing an application programming interface to processes hosted on computers of the distributed system(see col.10 lines 17 – 45);
passing a first message from a first process to a second process hosted on one computer of the distributed system, including passing said message through a shared memory accessible to both the first process and the second process (see col.10 lines 17 – 65) ; and

passing a second message from the first process to a third process hosted on a second computer of the distributed system, including passing said message over a communication channel coupling the first and the second computers (see col. 10 , lines 17 – 65). Dandrea does not explicitly teach an application programming interface, however Bruno teaches an application programming interface that provides an efficient generation and control or resource reservation on an operating system of a computer system, It would have been obvious to one of the ordinary skill in the art to include Application programming interface (API) in Dandrea invention because doing so would allow the language and the message format used by an application program to communicate with a program that provides services for it.

As to claim 26, Dandrea teaches the method of claim 22 wherein the first process uses the same application programming interface to pass the first message and the second message (see col. 10 lines 17 – 65).

As to claim 27, Dandrea teaches the method of claim 22 wherein the first process is unaware of whether the first message and the second message are passing to a process hosted on the first computer or the second computer (see col. 10 lines 17 – 65).

As to claim 28, Dandrea teaches the method of claim 22 wherein providing the application programming interface includes providing a queuing interface for passing messages between computers (see col. 10 lines 17 – 65).

As to claim 29, Dandrea teaches the method of claim 22 further comprising: providing a macro queue associated with the plurality of queues (see col. 4 lines 33 – 47); and

wherein passing the first message from the first process to the second process includes calling the application programming interface to initiate a request to the macro queue to obtain a message stored in one of the plurality of queues without identifying a particular queue and selecting a queue from among the plurality of queues and selecting a message from the selected queue(see col. 5 lines 14 – 40).

As to claim 30, Dandrea teaches the method of claim 22 further comprising: providing a remote queue proxy for establishing the communication channel between the first and the second computers (see fig. 1).

As to claim 31, Dandrea teaches a method for message passing in a distributed system comprising:

providing a queue manager on each of a group of computers in the distributed system (see col. 4 lines 33 – 47);

providing an application programming interface to processes on each of the computers of the group, including providing an interface to accept and to provide messages for passing between processes hosted on the computers (see col. 10 line 17 – 45);

collecting operational statistics at each of the queue managers related to passing of messages between processes using the application programming interface; and optimizing passing of the messages according to the collected statistics (see col. 4

line 48 – col. 5 line 14). Dandrea does not explicitly teach an application programming interface, however Bruno teaches an application programming interface that provides an efficient generation and control or resource reservation on an operating system of a computer system. It would have been obvious to one of the ordinary skill in the art to include Application programming interface (API) in Dandrea invention because doing so would allow the language and the message format used by an application program to communicate with a program that provides services for it.

As to claim 33, Dandrea teaches the method of claim 22 wherein enqueueing the message into each of the message queues includes performing a logically atomic enqueueing operation on all the queues (see col. 4 lines 48 – col. 5 line 14).

As to claim 34, Dandrea teaches the method of claim 22 wherein providing each of said replicated queues includes providing a replicated macro queue associated with a plurality of replicated member queues of said macro queue (see col. 4 lines 33 – 47).

As to claim 35, Dandrea teaches a method of managing messages, comprising: providing an application programming interface (API) to allow a producer module to send a message to a macro queue that manages a plurality of member queues, the API sending the message to the macro queue without identifying one of the plurality of member queues; and using the same API to allow the producer module to send a message to an individual queue (see col. 4 lines 33 – 65). Dandrea does not explicitly teach an application programming interface, however Bruno teaches an application programming interface that provides an efficient generation and control or resource

reservation on an operating system of a computer system, It would have been obvious to one of the ordinary skill in the art to include Application programming interface (API) in Dandrea invention because doing so would allow the language and the message format used by an application program to communicate with a program that provides services for it.

As to claim 36, Dandrea teaches the method of claim 35 wherein the macro queue selects one or more of the member queues according to a predefined criteria (see col. 5 line 9 – 40).

As to claim 37, Dandrea teaches the method of claim 36 wherein the macro queue, the member queues, and the individual queue are implemented as software objects according to object oriented programming principles (see col. 6 lines 25 – 40).

As to claim 38, Dandrea teaches a method of managing messages, comprising: providing an application programming interface (API) to allow a consumer module to retrieve a message from a macro queue that manages a plurality of member queues, the API retrieving the message from the macro queue without identifying one of the plurality of member queues (see col. 4 lines 33 – 47); and using the same API to allow the consumer module to retrieve a message from an individual queue (see col. 2 line 47 – col. 3 line 10).

As to claim 39, Dandrea teaches the method of claim 38 wherein the macro queue selects one of the member queues according to a predefined criteria and selects a message from the selected member queue (see col. 4 line 48 – col. 5 line 40).

As to claim 40, Dandrea teaches the method of claim 39 wherein the macro queue, the member queues, and the individual queue are implemented as software objects according to object oriented programming principles (see col. 6 lines 25 – 40).

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sargon N. Nano whose telephone number is (571) 272-4007. The examiner can normally be reached on 8 hour.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571) 272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sargon Nano
Jan.12, 2006



ARIO ETIENNE
PRIMARY EXAMINER